CHAPTER 12

PC Card

This chapter presents requirements and recommendations for PC Card. This includes 16-bit PC Card, previously referred to as Personal Computer Memory Card International Association (PCMCIA) cards, CardBus cards, and PC Card socket controllers.

Windows 98 and Windows NT support 16-bit PC Card I/O cards and CardBus I/O cards. Memory 16-bit PC Card cards are supported only as legacy devices. For any PC Card device to work effectively with Windows or Windows NT, the manufacturer must implement a minimum set of tuples documented in the PC Card standard. Windows uses these tuples to identify and configure any 16-bit PC Card card, and it might also use these tuples for CardBus cards.

Contents

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

PC Card Basic Requirements

This section summarizes the basic requirements for PC Card.

12.1. All devices comply with the PC Card standards

Required

Designs for PC Card socket controllers and cards must all be based on the PC Card standards.

The *February 1995 PC Card Standard* added requirements for minimum card information structure (CIS), 3.3-volt cards, multifunction cards, and cards that use DMA. It also introduced requirements for CardBus (32-bit) cards. The *March 1997 PC Card Standard* incorporated all corrections, changes, and adopted proposals as of that date.

All PC Card devices must comply with these standards for PC 99.

12.2. System and ZV-compatible 16-bit PC Cards comply with ZV standard definitions

Required

The PC Card standards define the requirements for Zoomed Video (ZV) cards and system support.

PC Card Socket Controller Requirements

This section summarizes requirements and standards for socket controllers.

12.3. Controller supports industry-standard ExCA register set

Required

The built-in software supporting 16-bit PC Card cards in Windows includes drivers for the industry-standard Exchangeable Card Architecture-compatible (ExCA-compatible) socket controllers. To be compatible with these drivers, socket-controller implementations must support the industry-standard ExCA base register set.

Notice that some controllers do not fully implement the register set and therefore are incompatible. Also, some controllers implement extended registers or enhancements. The built-in Windows drivers do not exploit these features, even though the controller might be compatible.

12.4. System maintains mapping of IRQ Routing Register bits to system interrupt vectors

Required

The system design must maintain the mapping of the PC Card controller's IRQ Routing Register bits to system interrupt vectors. This means that when an interrupt is programmed in the controller to occur on the IRQ*x* pin, the system's IRQ routing causes the interrupt controller to generate the interrupt vector for IRQ*x* and no other IRQ.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

12.5. IRQ connections can be determined by using the 0805 register

Required

Windows uses the 0805 register on CardBus controllers to determine which ISA IRQs are connected to the controller. This register must engage (drive low) the corresponding ISA IRQ when programmed with a value. It must disengage the IRQ (float high) when programmed at zero (0). This behavior must be achieved without requiring the operating system to program any non-standard registers.

12.6. CardBus controllers support both ISA and PCI interrupts

Required

PC Card software dynamically configures the bridge to use ISA interrupts for 16-bit PC Card cards and to use Peripheral Component Interface (PCI) interrupts for CardBus cards. As defined in requirement 12.5, “IRQ connections can be determined by using the 0805 register,” and requirement 12.4, “System maintains mapping of IRQ Routing Register bits to system interrupt vectors,” CardBus controllers must maintain mapping of IRQ routing. Also, notice that systems implementing CardBus controllers must fully support PCI 2.1 as well as additional PCI requirements for IRQ routing as described in requirement 9.13, “Interrupt routing is supported using ACPI,” and requirement 9.14, “BIOS does not configure I/O systems to share PCI interrupts.”

12.7. System supports industry-standard definition for CardBus bridges

Required

Systems must support the definition in *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification) for CardBus controllers (PCI-to-CardBus bridges). This definition includes a common PCI Configuration Space header assigned the Header Type field value of 82h.

Although this requirement is not yet incorporated into the PCI standard, Windows supports it. Any controller features that are not part of the Yenta specification will not be used in standard drivers. The BIOS is responsible for any hardware initialization or setup required to make the controller comply with the Yenta specification or other requirements listed in this chapter.

Because CardBus host controllers are PCI bus bridges, they will be supported (enumerated and configured) by the PCI software in Windows in the same manner as other PCI bus bridges. For more information, see requirement 9.3, “System uses standard method to close BAR windows on nonsubtractive decode PCI bridges.”

12.8. BIOS initializes CardBus controller in 82365-compatible mode and supports backward compatibility

Recommended

CardBus controllers are enumerated and configured in the same way as other PCI bus bridges. The PCI bus bridge support in Windows 98 is based on requirements for PCI interrupt routing and bridge-window configuration. Because of this, full compliance with the latest PCI specifications is a requirement for CardBus support. See Chapter 9, “PCI.”

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

There are steps the BIOS can take to achieve backward compatibility with Windows. Specifically, the BIOS can initialize the CardBus controller in Intel 82365-compatible mode and report it as device “PNP0E03, Intel 82365-compatible CardBus controller.” The requirements are as follows for BIOS POST time (CardBus controller ConfigSpace initialization):

- Command register (offset 0x04) set to 0x07 (IOSpaceEnable, MemSpaceEnable, BusMasterEnable).
- RegisterBaseAddress (offset 0x10) set to 0. If support for other environments is needed, such as Windows 3.1 or MS-DOS, some other value can be set.
- All memory and I/O windows (offset 0x1c–0x38) set to 0.
- Interrupt Line register (offset 0x3c) set to 0xff (no IRQ assigned). If support for other environments is needed, such as Windows 3.1 or MS-DOS, an assigned IRQ line can be set. Notice, however, that this register must be set to 0xff at the time that the device is disabled by the operating system, and then set into CardBus mode. More information about BIOS enumeration is presented later in this requirement.
- Other controller-specific initialization as required to put the controller in legacy mode.

This puts the CardBus controller into legacy mode where the Windows Socket Services driver can access it as an Intel PC Card I/O card–compatible (PCIC-compatible) controller at an I/O address, for example, 0x3e0.

Notice that the BIOS must be at least PCI 2.1-compliant and must support the $PIR Interrupt Routing Table. The $PIR table must return the necessary PCI IRQ routing information, including the routing information for the CardBus controller. In general, if the CardBus controller is on the system board, there must be a slot routing entry for it in the table. If the CardBus controller is a PCI add-on card, there must be routing information entries for each PCI slot in the system.

During Plug and Play BIOS enumeration, the BIOS should report the CardBus controller as *pnp0e03 with a compatible ID of *pnp0e00 and the I/O resource of two ports, for example, 0x3e0–0x3e1.

For more information, see the white paper on CardBus host controllers and Windows compatibility at http://www.microsoft.com/hwdev/busbios/.

12.9. CardBus controllers do not share writable PCI Configuration Space bits

Required

CardBus controllers are multifunction PCI devices, and Windows treats each function as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits, such as the Command register.

Notice that the 16-bit PC Card interface legacy-mode BAR (offset 44h in the Type 2 PCI header) is the only exception to this requirement. This BAR must be shared between the two functions in order to be compatible with the ExCA programming model.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

12.10. Each 16-bit PC Card memory window in CardBus controller has it own page register

Required

For complete flexibility and support of typical configurations, CardBus controllers must support the independent location of R2 memory windows anywhere in the full system address space as recommended in the Yenta specification.

Controllers that share a single page register among all 16-bit PC Card memory windows require that all 16-bit PC Card memory windows must be located within the same 16-MB block. Often, this is not possible with typical (16 MB) DRAM and bridge (positive-decode) configurations. The result is disabled cards.

Plug and Play Design for 16-bit PC Card Cards

This section summarizes the Plug and Play requirements for 16-bit PC Card cards.

The Windows operating system determines what type of card is plugged into the PC Card socket by examining the tuples on the card. For Plug and Play functionality, 16-bit PC Card I/O cards must support a set of required information and configuration tuples. The PCMCIA bus enumerator uses these tuples to identify the card, load the correct device driver, and indicate all possible configurations to the Plug and Play configuration manager. The operating system then dynamically assigns a valid configuration based on this information.

12.11. Card supports required I/O card tuples

Required

The following items must be implemented for any 16-bit PC Card I/O card that connects to a PC 99 system:

- The 16-bit PC Card card must contain:
- The device information tuple (CISTPL_DEVICE, 01h for cards capable of 5V operation or CISTPL_DEVICE_0C, 1Ch for cards capable of 3.3V operation).
- The Level 1 (L1) version/product information tuple (CISTPL_VERS_1, 15h).
- The configuration tuple (CISTPL_CONFIG, 1Ah).
- The configuration table entry tuple (CISTPL_CFTABLE_ENTRY, 1Bh).
- A 16-bit PC Card card with more than 64 MB Common Memory must contain the extended device information tuple (CISTPL_EXTDEVICE, 09h).
- The L1 version/product information tuple must contain the product name and manufacturer name in the product information string (TPLLV1_INFO, byte 4).
- The product name and manufacturer name in the L1 version/product information tuple must be composed only of ASCII characters greater than ASCII 20h and less than ASCII 7Fh.

Windows uses the information contained in the required and recommended tuples to create a unique device ID for the card and to assimilate configuration information for the device. Windows uses the device configuration tuples to determine the general characteristics of the card.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

Required I/O Card Tuples

Tuple ID	Tuple code	Description and comments
01h	CISTPL_DEVICE	Device information (common memory). For non-memory cards, this tuple must be present, but the device type will be NULL.
15h	CISTPL_VERS_1	L1 version/product information strings: Product information Product name Product number Other manufacturer information
1Ah	CISTPL_CONF	Configuration. Indicates the location of configuration registers and registers present.
1Bh	CISTPL_CE	Configuration table entry. Appropriate configuration requirements for I/O space, interrupts, memory, and so on should be specified.
20h	CISTPL_MANFID	Manufacturer ID. Card manufacturer ID code. Defines manufacturer for this card.
21h	CISTPL_FUNCID	Function ID. Provides function information about the card. Also includes system initialization information.

The device information tuple provides information about the memory devices used in the card's common memory space. The device type, size, and speed are used to configure the socket for efficient access to the card. This tuple must be present on 16-bit PC Card I/O cards, but the device type must be NULL.

The L1 version/product information tuple contains human-readable information about the product and its manufacturer. This information is intended to be displayed to the user where necessary. Windows uses the information contained in the product information string and product name string to construct the device ID for that card. It also scans through the tuple, starting at the very beginning and continuing to the end of the product name string.

The information gathered from the L1 version/product information tuple is used to construct the unique device ID. Because the optional third and fourth strings in the tuple are not used in the unique ID, devices that require unique numbers on each card can use these strings to store that information.

The configuration tuple tells the software where to locate the configuration registers that program the card's configuration, as well as which registers are present on the card.

Each configuration table entry tuple completely describes one valid configuration in which the card can operate. Each entry describes power, timing, I/O space, interrupt, and memory space requirements for the given configuration. Configuration software selects one of these configurations for the card based on the resources currently available in the system.

The manufacturer ID tuple (CISTPL_MANFID, 20h) and the function ID tuple (CISTPL_FUNCID, 21h) add extra flexibility to a PC Card that connects to the PC:

- The manufacturer ID tuple provides unique information about the card manufacturer. This code is registered with PCMCIA. Windows uses the manufacturer ID tuple as one source for creating a 16-bit CRC used in the construction of the device ID.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

- The function ID tuple provides information about the class of device or what function the card provides, for example, memory, modem, disk, and so on. This information helps the software perform necessary installation tasks and locate compatible drivers. Although it is not required to make this determination, Windows uses the function ID tuple internally to determine what type of device is on the PC Card.

12.12. Configuration table entry tuples listed in priority order

Required

Configuration table entry tuples are placed in the preferred order for configuring the device. Windows processes the tuples in the order they are placed in the Card Information Structure (CIS). From these tuples, Windows creates a logical configuration in this order and prioritizes them in the same order. Notice that for multiple voltage cards, the voltage policy is to prioritize 3.3-volt configurations, if they are supported by the system, over 5-volt configurations, regardless of the order of the configuration table entry tuples (CISTPL_CFTABLE_ENTRY).

12.13. Card specifies maximum configuration options

Required

Many older PC Cards specified fixed configurations in order to address compatibility with existing software. However, this is not the intended use for tuples; the configuration software should be responsible for compatibility. The tuples should be used only to describe its maximum configurability, ruling out configurations not supported by the hardware.

If fixed configurations must be provided for an operating system other than Windows, there must be one or more entries that specify the maximum configurability that the hardware can handle. An example of "maximum configurability" is to specify "any IRQ" rather than only IRQ 3 or IRQ 4.

Plug and Play Design for CardBus

This section summarizes the Plug and Play requirements for CardBus cards. CardBus was designed as a combination of the 16-bit PC Card and PCI. The goal is to gain the benefits of PCI in a PC Card format. Consistent with this goal, Windows support for CardBus places specific requirements on CardBus cards.

12.14. Configuration Space meets Common Silicon Guidelines

Required

The Common Silicon Guidelines are defined in Section 2.6 of the *PC Card Standard Guidelines, Volume 10.*

The standard for CardBus defines a PCI-like Configuration Space that is not fully compliant with the PCI specification. Specifically, under the CardBus standard, card vendors do not have to implement certain critical fields in the Configuration Space, described in the PC Card standard as allocated. In the PC Card standard guidelines for silicon common to both PCI and CardBus products, the implementation of these fields is recommended.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

However, to maintain compatibility with existing PCI system software and drivers for PC 99, Windows will support only CardBus cards whose Configuration Space is designed to meet the Common Silicon Guidelines. This is a requirement because CardBus configuration is performed by the PCI software, which can deal with all aspects of PCI topology configuration, including bridging. Without the allocated fields, the cards cannot be fully treated as PCI devices and cannot be supported under Windows.

The required allocated fields are listed in the following table.

Required Allocated Fields

Field	Description and comments
Vendor ID	This read-only field contains a unique ID (in PCI space) for the card manufacturer. The PCI SIG allocates unique IDs.
Device ID Revision ID	These read-only fields are vendor-assigned values that uniquely identify the device (among all vendors of PCI or CardBus products).
Class Code	This read-only field is defined in PCI 2.1. It describes what type of device the card is.
Max_Lat Min_Gnt	These read-only fields specify the desired settings for Latency Timer values according to PCI 2.1. A value of 0 indicates the device has no major requirements for the settings of Latency Timers.
Interrupt Line	This register must be read-write and must not be connected to anything, just as on PCI cards. This register is used to store the current IRQ routing for the device.

12.15. RESERVED fields comply with PCI 2.1

Required

The CardBus specification also lists two RESERVED fields (offset 2C in the Configuration Space), which have since been defined in PCI 2.1. These fields are also required on CardBus cards for Windows compatibility.

Required RESERVED Fields

Field	Description
Subsystem ID	If different from Device ID
Subsystem Vendor ID	If different from Vendor ID

12.16. CardBus card implements required and recommended tuples

Required

For CardBus, Windows also requires the same set of card tuples recommended in the PC Card guidelines, as summarized in the following table.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

Required CardBus Tuples

Tuple ID	Tuple code	Comments
04h	CISTPL_CONFIG_CB	—
05h	CISTPL_CFTABLE_ENTRY_CB	—
07h	CISTPL_BAR	—
13h	CISTPL_LINKTARGET	Required as first tuple in PC Card standard.
15h	CISTPL_VERS_1	—
20h	CISTPL_MANFID	—
FFh	CISTPL_END	Required as end-of-chain tuple in PC Card standard.
21h	CISTPL_FUNCID	Recommended in PC Card standard; required for PC 99.

PC 99 Requirements for PC Card

This section summarizes additional requirements for PC Card.

Power Management for PC Card

This section summarizes the specific power management requirements for PC Card. Power management requirements for specific device classes are defined in the related chapters in Part 4 of this guide.

12.17. Socket controller complies with device class power management reference specification

Required

This applies for both 16-bit PC Card-only controllers and CardBus controllers.

The *PC Card Controller Device Class Power Management Reference Specification, Version 1.0* or later, provides class-specific definitions of the OnNow device power states (D0–D3) for these devices. The specification also covers device functionality expected in each power state and the possible wake-up event definitions for the class, for example, whether card insertion should wake the system.

12.18. 16-bit PC Card cards implement power-related events using ReqAttn bit and #STSCHG mechanism

Required

Any 16-bit PC Card card that is capable of signaling a wake-up event to the system, as defined in the device class power management reference specification for its class, must implement the ReqAttn bit and its associated enable bit in the Extended Status register, and must signal on the #STSCHG line.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

12.19. CardBus controllers and cards implement PCI power management specifications

Required

PCI-to-CardBus bridges must comply with the requirements defined in *PCI Bus Power Management Interface Specification, Revision 1.0* or later. CardBus cards must also comply with the requirements defined in *PCI Bus Power Management Interface Specification, Revision 1.1* or later.

The CardBus card must use the CSTSCHG pin to signal wake-up events. This is because there is no PME# pin on the CardBus interface, and the CardBus card must use PME_EN in the card's Configuration Space to enable wake-up events. Specifically, setting the PME_EN bit in the card's Configuration Space must provide the same behavior as setting both the GWAK and WKUP bits in the card's Function Event Mask register.

For more information, see "Power Management for PCI Controllers and Peripherals" in Chapter 9, "PCI."

Device Drivers and Installation for PC Card

This section summarizes requirements for PC Card device drivers.

12.20. No user intervention required for correctly installing devices

Required

The user must not be required to perform any device-installation action other than to insert disks that contain drivers and other files.

12.21. Device is immediately functional without restarting the system

Required

The user must be able to begin using the device without having to restart the system. Device use begins either after installation is complete or whenever the device is inserted in the system.

12.22. ZV-compatible PC Card driver uses DirectDraw LVE

Required

ZV-compatible PC Card drivers must use software interfaces based on 32-bit DirectDraw Live Video Extensions (LVE) in order to configure the graphics controller to receive video input using the ZV port. This includes programming the graphics controller to configure the format of the video data, its location on screen, and so on. LVE is part of DirectX 3.0.

ZV card device drivers must handle dynamic graphics state changes, such as resolution changes, color depth changes, and switching to and from full-screen MS-DOS–based applications.

12.23. 16-bit PC Card card driver supports sharing of level-mode interrupts

Required

CardBus systems support both 16-bit PC Card cards and CardBus cards. In this environment, interrupt sharing becomes an issue because CardBus controllers can use PCI interrupts, which are level-sensitive and sharable. To help alleviate interrupt limitations in CardBus systems, Windows operating systems can take advantage of PCI interrupt-sharing capabilities.

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

In cases where no ISA IRQs are available to a 16-bit PC Card card in a CardBus controller, the operating system will assign a PCI interrupt to the card. Therefore, it is recommended that 16-bit PC Card card drivers are updated to support interrupt sharing. However, it is a requirement that 16-bit PC Card card drivers must "hook" the interrupt (whether sharable or not) before its hardware generates any interrupts.

See also requirement 9.15, "BIOS configures boot device IRQ and writes to the interrupt line register."

PC Card References

The following represents some of the references, services, and tools available to help build hardware that is optimized to work with Windows operating systems.

CardBus host controllers and Windows compatibility white paper
http://www.microsoft.com/hwdev/busbios/

Microsoft Windows 95 DDK and Windows NT 5.0 DDK
MSDN Professional membership

PC Card Controller Device Class Power Management Reference Specification, Version 1.0
http://www.microsoft.com/hwdev/onnow.htm

PC Card diagnostic utility (Dtpl.exe)and white papers
http://www.microsoft.com/hwdev/busbios/.

PCI Bus Power Management Interface Specification, Revision 1.0 and later
PCI Local Bus Specification, Revision 2.1 (PCI 2.1)
http://www.pcisig.com

PCMCIA standards, including *PC Card Standard Guidelines* and *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification)
PCMCIA
2635 North First Street, Suite 209
San Jose, CA 95134 USA
Phone: (408) 433-2273
Fax: (408) 433-9558
E-mail: office@pcmcia.org
http://www.pc-card.com/

Checklist for PC Card

If a recommended feature is implemented, it must meet the requirements for that feature as defined in this document.

12.1. All devices comply with the PC Card standards
Required

12.2. System and ZV-compatible 16-bit PC Cards comply with ZV standard definitions
Required

12.3. Controller supports industry-standard ExCA register set
Required

12.4. System maintains mapping of IRQ Routing Register bits to system interrupt vectors
Required

12.5. IRQ connections can be determined by using the 0805 register
Required

Published: June 15, 1998. Printed: 06/24/98 1:16 PM

12.6. CardBus controllers support both ISA and PCI interrupts
Required

12.7. System supports industry-standard definition for CardBus bridges
Required

12.8. BIOS initializes CardBus controller in 82365-compatible mode and supports backward compatibility
Recommended

12.9. CardBus controllers do not share writable PCI Configuration Space bits
Required

12.10. Each 16-bit PC Card memory window in CardBus controller has it own page register
Required

12.11. Card supports required I/O card tuples
Required

12.12. Configuration table entry tuples listed in priority order
Required

12.13. Card specifies maximum configuration options
Required

12.14. Configuration Space meets Common Silicon Guidelines
Required

12.15. RESERVED fields comply with PCI 2.1
Required

12.16. CardBus card implements required and recommended tuples
Required

12.17. Socket controller complies with device class power management reference specification
Required

12.18. 16-bit PC Card cards implement power-related events using ReqAttn bit and #STSCHG mechanism
Required

12.19. CardBus controllers and cards implement PCI power management specifications
Required

12.20. No user intervention required for correctly installing devices
Required

12.21. Device is immediately functional without restarting the system
Required

12.22. ZV-compatible PC Card driver uses DirectDraw LVE
Required

12.23. 16-bit PC Card card driver supports sharing of level-mode interrupts
Required

Published: June 15, 1998. Printed: 06/24/98 1:16 PM